Sampling Rate Optimization for IEEE 802.11 Wireless Control Systems

Dohwan Kim
DGIST
Daegu, Republic of Korea
dhkim@dgist.ac.kr

Yuchang Won
DGIST
Daegu, Republic of Korea
yuchang@dgist.ac.kr

Seunghyeon Kim
DGIST
Daegu, Republic of Korea
seu7704@dgist.ac.kr

Yongsoon Eun
DGIST
Daegu, Republic of Korea
yeun@dgist.ac.kr

Kyung-Joon Park*
DGIST
Daegu, Republic of Korea
kjp@dgist.ac.kr

Karl H. Johansson
KTH Royal Institute of Technology
Stockholm, Sweden
kallej@kth.se

ABSTRACT

Design of wireless control systems has been extensively studied, which is one of the fundamental issues in cyber-physical systems. In this paper, we *empirically* investigate *heterogeneous* sampling rate assignment with a testbed when multiple physical systems are controlled through an IEEE 802.11 network. Among the critical design variables in wireless control systems, we focus on the sampling rates because they are always key control knobs regardless of network protocols. There has been *little experimental research* on heterogeneous sampling rate optimization for IEEE 802.11 wireless control systems, where the sampling rates of each control loop may have different values. We first formulate the co-design problem in an optimization framework with respect to the heterogeneous sampling rates by explicitly taking into account the relations of the sampling rates with the control cost, network energy consumption, and network delay. We further relax the problem as convex optimization, which is provably solved in polynomial time. Our empirical study ensures that the approximate solution is tightly close to the original optimum. To validate the proposed optimization framework, we build a disk-levitation tube testbed, which wirelessly controls the height of 20 disks at the same time. Our empirical study confirms that our optimization formulation is highly effective in practice.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; *Sensor networks*; • **Theory of computation** → *Mathematical optimization*.

---

*K.-J. Park is the corresponding author.

KEYWORDS

Cyber-physical systems, wireless control systems, IEEE 802.11, sampling rate assignment

1 INTRODUCTION

Generally speaking, a cyber-physical system (CPS) is an integration of computation with physical processes by networking [7, 10, 19]. There is a growing interest in design of CPS by wireless networks, which have many advantages over wireline such as reduction of deployment and maintenance cost, enhanced flexibility, expandability, and mobility. However, at the same time, the introduction of wireless networks raises new challenges for design of CPS. Wireless channels have adverse effects of path loss, multi-path fading, and interference, which may result in packet loss. Furthermore, the shared nature of the wireless channel requires a deliberate network and control co-design for real-time control applications.

Wireless control systems has been extensively studied so far [14]. From the perspective of network protocols, we can categorize most of related work into three cases: *(i)* Details of protocols are not explicitly incorporated while the network-induced delay and packet loss are modeled by certain random variables. *(ii)* Time-division multiple access (TDMA) is considered for guaranteeing timely delivery of packets, which incurs little analytic difficulty. *(iii)* Random access networks are adopted for their simple and distributed nature in practice, while they introduce significant analytic difficulty. Among these three cases, *(iii)* is less investigated compared to *(i)* and *(ii)*.

From the perspective of system design, the two key control variables are the *sampling rates* and the *network parameters*. Control performance initially improves with higher sampling rates. However, the network delay will also increase with higher sampling rates because of increased network traffic and at a certain point, control performance will degrade due to severe network delay. Hence, the sampling rates not only have explicit functional relations with

control performance, but also have implicit ones through the interaction with the network. In the meanwhile, the network-induced delay and packet loss are also functions of the network parameters. However, the actual network parameters vary among protocols. Here, we focus on the sampling rates because they are always key control knobs regardless of network protocols.

In this paper, we investigate how to set the sampling rates of each control loop when multiple physical systems are controlled through an IEEE 802.11 network. So far, there has been little *experimental* work on *heterogeneous* sampling rate assignment for IEEE 802.11 wireless control systems, where the sampling rates of each control loop may have different values. Our contributions can be summarized as follows:

- We formulate the sampling rate assignment for IEEE 802.11 wireless control systems in an optimization framework, where the relation of the heterogeneous sampling rates with the control cost, network energy consumption, and network delay are incorporated.
- We further relax the formulated optimization to derive a convex optimization problem, which gives a provable optimal solution in polynomial time. Our performance evaluation confirms that the approximate solution is tightly close to the original optimum while significantly reducing computation time.
- We *empirically* validate the proposed optimization framework with a disk-levitation tube testbed, which wirelessly controls the height of 20 disks at the same time. Our experimental results show the practical effectiveness of the proposed framework.

The remainder of the paper is organized as follows: Related work is given in Section 2. The system model and the problem formulation is described in Section 3. We evaluate the performance of the proposed optimization framework in Section 4. Finally, our conclusions follow in Section 5.

2 RELATED WORK

There have been extensive studies on wireless control systems. Consequently, it is formidable to provide a thorough introduction in one section. A comprehensive survey can be found in [14]. Here, we introduce recent work categorized by three cases of network protocol consideration in Section 1.

In [5], co-design of packet forwarding policies and controllers is studied in multi-hop wireless networks. The objective of co-design is to schedule transmission for optimal control under packet loss with a stochastic network model. The problem of sharing a wireless channel between multiple control loops is studied in [6]. The goal is to guarantee that the wireless interference does not violate stability of control loops. To this end, a sufficient mathematical condition for sensor access policies is derived.

An optimization problem for WireleeHART is formulated in [17] to minimize a control cost under delay constraints. Then, the optimization is further approximated as a convex one. Multiple algorithms such as greedy heuristic, simulated annealing, and sub-gradient method are used to solve the original and approximate optimization. A cross-layer optimized control (CLOC) protocol is proposed in [13] for minimizing the worst case performance loss of

multiple industrial control systems in IEEE 802.15.4 The constraints include scheduling, routing, and link congestion. The control variables are sampling rates, the load distribution of routing, and the weight of concurrent transmission sets. A joint optimization framework for control and communication in TDMA is provided in [16]. The constraints incorporate stochastic form of the maximum allowable time interval and maximum allowable delay. An optimization formulation is presented in [2] for minimizing the tracking error due to discretization, packet delay, and loss in multi-hop wireless networks, which is modeled as a directed graph. In [8], the authors present integration of networked control design and realistic wireless mesh networks based on WirelessHART that explores the interactions between wireless routing and control design.

Duty-cycle optimization is proposed in [15] for IEEE 802.15.4 random access networks to minimize power consumption while ensuring the reliability and delay requirements. The duty-cycle optimization problem finds the optimal sleep time and listening time of each receiver node that the overall power of the network is minimized. In [4], a Just-in-Time LEarning-based Adaptive Parameter tuning (JITLEAP) algorithm is proposed that adapts the CSMA/CA parameter setting to the time-varying operating conditions by exploiting the past history to find the most appropriate setting for the current conditions. A co-design framework of 802.15.4 CSMA/CA and controllers is proposed in [11]. The objective is to minimize the network energy consumption the constraints are the packet loss probability and delay. The constrained optimization problem finds the homogeneous optimal sampling period and the network parameters. An optimization problem is proposed in [12] for minimizing energy consumption of the network subject to reliability and delay constraints that gives the optimal MAC parameters.

In summary, in spite of extensive related work, *experimental research on heterogeneous sampling rate optimization* has not been properly carried out in IEEE 802.11 wireless control systems.

3 PROBLEM FORMULATION

3.1 System Model

We consider a wireless networked control system, where the physical systems periodically send sensed data to the controllers over a wireless network. The overall system architecture is shown in Fig. 1. Here, we assume that actuators are directly attached to the controllers while sensors are wirelessly connected to the controllers as in many related studies [16]. In fact, this architecture is practically meaningful in numerous industrial applications, where sensing and actuation points are apart from each other [3].

The system consists of N control loops, where each physical system and controller are respectively denoted by P_i and C_i, $i = 1, 2, \cdots, N$. Here, we consider periodic sampling of each physical system P_i with a sampling period of h_i, $i = 1, 2, \cdots, N$. Then, each sensed data is sampled and transmitted to the controller C_i at a sampling rate of $\lambda_i = 1/h_i$ Hz. We consider IEEE 802.11 CSMA/CA or so-called Wi-Fi for random access. It should be noted that other random access networks such as IEEE 802.15.4 CSMA/CA can be considered in a similar manner by using different parameter values.

The medium access control (MAC) protocol of IEEE 802.11 operates as follows: A tagged node that intends to transmit first senses the channel. It delays its transmission if the channel is busy. If
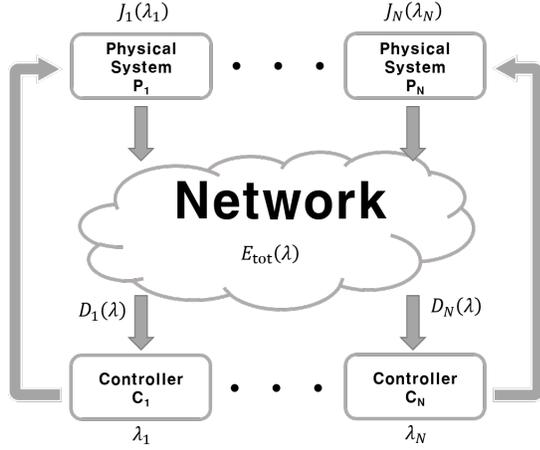
Figure 1: Overview of the wireless control system setup.

the channel is sensed idle for a specific time interval, the node chooses a random back-off timer, which is uniformly distributed in $[0, CW - 1]$, where CW is called the contention window size of the node. Here, CW is initially set to its minimum value $CW_{\min}$, and is doubled up to its maximum value of $CW_{\max}$ after each collision. The back-off timer is decreased by one if the channel is sensed idle for one physical slot time, suspended if the channel is busy, and reactivated after the channel is sensed idle again. The node transmits its packet when the back-off timer reaches zero. After the data packet is received without errors, the receiver sends an acknowledgment packet to the sender. If the acknowledgment is not received, the data packet is presumed to be lost, and a retransmission is scheduled. Retransmissions for the same data packet can be made up to a pre-determined retry limit. Beyond this limit, the pending packet will be dropped. The CW value is reset to $CW_{\min}$ when a packet has been successfully transmitted.

For each control loop, the control cost is denoted by J_i, $i = 1, \cdots, N$, which is an the performance index for the tracking deviation of the controller. Similarly as in [9, 17], the control cost J_i is calculated as follows:

$$J_i = w_{c_i} \alpha_i e^{-\beta_i \cdot \lambda_i} + C_i, \quad (1)$$

where w_{c_i} is the weight for the i-th control loop, α_i is the magnitude coefficient, β_i is the decay rate, and C_i is a constant. We determine these coefficients by curve fitting with the empirical data from our testbed in the subsequent section.

For network energy consumption, the energy used for transmission at the sensors accounts for the major portion, which is also responsible for sensor lifetime because batteries are typically used for sensors. Hence, the energy consumption of the network is expressed as a function of the sampling rate λ_i and the collision probability p_i of the nodes as follows:

$$E_{\text{tot}} = \sum_{i=1}^{N} \lambda_i \left(E_S + \frac{p_i}{1 - p_i} E_C \right), \quad (2)$$

where E_S and E_C is the energy consumed for each successful transmission and collision, respectively.

3.2 Network and Control Co-Design

In the traditional digital control, a higher sampling rate improves control performance though the performance saturates at a certain point. In the meantime, the performance of networked control initially improves with a higher sampling rate. However, since network traffic is proportional to the sampling rate, more contention will occur with a higher sampling period. Consequently, the network delay will increase and at a certain point, the deadline for packet delivery will be missed, which will degrade the control performance and even cause instability. Consequently, unlike traditional digital control, the sampling rate should be more carefully designed for wireless control systems.

Here, the goal of co-design is to minimize the weighted sum of the control cost in (1) and the network energy consumption in (2) with λ_i's as the control variables. As a constraint, the total time spent for successful transmission of each packet should not exceed the sampling period. In addition, each control loop should maintain its sampling rate in the range of the minimum and the maximum values. Then, the overall optimization can be formulated as follows:

$$\underset{\lambda}{\text{minimize}} \quad F(\lambda) = W \cdot \sum_{i=1}^{N} J_i(\lambda_i) + (1 - W) \cdot E_{\text{tot}}(\lambda) \quad (3)$$

$$\text{subject to} \quad D_i(\lambda) < \frac{1}{\lambda_i}, \ i = 1, \cdots, N$$

$$\lambda_{\min, i} \leq \lambda_i \leq \lambda_{\max, i}, \ i = 1, \cdots, N.$$

Here, W ($0 \leq W \leq 1$) is a weight between the control cost and the energy consumption, and the network delay D_i is the time spent for successful transmission of each packet, which is a function of the control variable $\lambda = (\lambda_1, \cdots, \lambda_N)$.

In the optimization problem of (3), the objective function is readily given by (1) and (2). The main challenge is to derive a formula for $D_i(\lambda)$. By following a similar line of derivation in [18], we express D_i as a function of $\lambda = (\lambda_1, \cdots, \lambda_N)$. In the network, the average backoff window of node i is given by

$$\overline{W}_i = \frac{1 - p_i - p_i (2p_i)^m}{1 - 2p_i} \cdot \frac{CW_{\min}}{2}, \quad (4)$$

where p_i is the collision probability of node i and m is the retry limit, i.e., the maximum number of retransmissions.

In the meantime, we can express p_i as follows:

$$p_i = 1 - \prod_{\substack{j=1 \\ j \neq i}}^{N} P_j [\text{SE}], \quad (5)$$

where $P_i [\text{SE}]$ denotes the probability that node i does not transmit in a slot. Here, $P_i [\text{SE}]$ is given by

$$P_i [\text{SE}] = 1 \cdot (1 - \rho_i) + \rho_i \cdot P_i [\text{SE}|\text{QNE}], \quad (6)$$

where QNE represents "queue not empty", and ρ_i denotes queue utilization of node i. $P_i [\text{SE}|\text{QNE}]$ is expressed as $\left(\overline{W}_i - 1 \right) / \overline{W}_i$. Thus, we have

$$P_i [\text{SE}] = (1 - \rho_i) + \rho_i \frac{\overline{W}_i - 1}{\overline{W}_i} = 1 - \frac{\rho_i}{\overline{W}_i}, \quad (7)$$

and the packet collision probability of node i is given by

$$p_i = 1 - \prod_{\substack{j=1 \\ j \neq i}}^{N} \left(1 - \rho_j \frac{1 - 2p_j}{1 - p_j - p_j(2p_j)^m} \frac{2}{CW_{\min}} \right). \quad (8)$$

Now, we characterize the average time to serve a packet from node i, which is $1/\mu_i$, where μ_i is the service rate of node i. Since the sampling rate λ_i acts as the arrival rate of node i, we can determine ρ_i once we obtain μ_i since $1/\mu_i = \rho_i/\lambda_i$. The service time $1/\mu_i$ is given as follows:

$$\frac{1}{\mu_i} = \sum_{\substack{j=1 \\ j \neq i}}^{N} \left[T_S + T_C \frac{p_j}{2(1 - p_j)} \right] + \overline{W}_i + T_S + T_C \frac{p_i}{2(1 - p_i)}. \quad (9)$$

Hence, we have

$$\rho_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_i \rho_j \left[T_S + T_C \frac{p_j}{2(1 - p_j)} \right] + \lambda_i \left[\overline{W}_i + T_S + T_C \frac{p_i}{2(1 - p_i)} \right]. \quad (10)$$

The probability that any of the backoff slots experienced by node i is active is as follows:

$$q_i = 1 - \prod_{\substack{j=1 \\ j \neq i}}^{N} \left(1 - \frac{\rho_j}{\overline{W}_j} \right). \quad (11)$$

Then, the probability that a slot results in a collision given that it is active is given by

$$q_{c_i} = P\left[\text{collision} \mid \text{slot active} \right]$$

$$= \frac{1 - \prod_{\substack{j=1 \\ j \neq i}}^{N} \left(1 - \frac{\rho_j}{\overline{W}_j} \right) - \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{\rho_j}{\overline{W}_j} \prod_{\substack{k=1 \\ k \neq j, i}}^{N} \left(1 - \frac{\rho_k}{\overline{W}_k} \right)}{1 - \prod_{\substack{j=1 \\ j \neq i}}^{N} \left(1 - \frac{\rho_j}{\overline{W}_j} \right)}. \quad (12)$$

By putting (4), (11), (12) all together, we can express the network delay D_i by using the following relation.

$$D_i = \delta \left[\overline{W}_i \left((q_i - q_{c_i}) \cdot T_S + q_{c_i} \cdot T_C + (1 - q_i) \cdot 1 \right) + T_S \right]. \quad (13)$$

Hence, we can now solve (3) since all the terms are expressed as a function of the control variable $\lambda = (\lambda_1, \cdots, \lambda_N)$. However, since (3) is highly nonlinear and non-convex due to (13), we can only obtain sub-optimal solutions. In addition, the computational burden of algorithms may become severe as the number of nodes increases.

3.3 Convex Relaxation

In this section, we further relax the optimization of (3) as a convex problem, whose optimal solution can be provably obtained in polynomial time. In (13), the relation of D_i with λ is not given in a closed form. Hence, we look into D_i from a different viewpoint as a service time of a tagged node.

In the network with multiple nodes, a tagged node can experience one of the four states when it tries to transmit; backoff procedure, busy channel, transmission, and collision. The average total time for the tagged node to transmit a packet should be less than the sampling period. Then, D_i consists of the average backoff time D_{B_i},

the average transmission time of other nodes D_{OS_i}, the average collision time of the tagged node D_{C_i}, and the average collision time of other nodes D_{OC_i}. Then, the delay constraint in (3) can be re-written as

$$D_i = D_{B_i} + D_{S_i} + D_{OS_i} + D_{C_i} + D_{OC_i} < \frac{1}{\lambda_i}. \quad (14)$$

The average backoff time of the tagged node i, D_{B_i}, can be expressed by the uniform random variable of $CW_{\min}$ and the number of packet collisions as follows:

$$D_{B_i} = \delta \cdot E\left[\mathcal{U}\{0, CW_{\min} - 1\} \right] \cdot \left(1 + \frac{p_i}{1 - p_i} \right), \quad (15)$$

where $\mathcal{U}\{\cdot\}$ is the discrete uniform distribution, and p_i is the collision probability in (8). The average transmission time D_{S_i} of the tagged node i during the sampling period of $1/\lambda_i$ is T_S. The average transmission time of other nodes D_{OS_i} can be expressed as s function of the sampling rates of the nodes as follows:

$$D_{OS_i} = \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{\lambda_j}{\lambda_i} \cdot T_S. \quad (16)$$

We further approximate p_i in (8) by $\hat{p}_i$ under the assumption that a node's transmission attempt is independent of other node's transmission attempts. The tagged node experiences a collision when at least one node except the tagged one tries to transmit. In this case, p_i can be approximated by $\hat{p}_i$ as follows:

$$\hat{p}_i = 1 - \prod_{\substack{j=1 \\ j \neq i}}^{N} \left(1 - \frac{\delta \cdot \lambda_j}{1 - \lambda_i \cdot D_{OS_i}} \right) \approx \frac{\delta \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j}{1 - \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j T_S}. \quad (17)$$

By putting (17) into (15), D_{B_i} is given as

$$D_{B_i} = \delta \cdot \frac{CW_{\min} - 1}{2} \cdot \left(1 + \frac{\delta \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j}{1 - (T_S + \delta) \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j} \right). \quad (18)$$

The average collision time of the tagged node i, D_{C_i}, can be expressed as the product of the time required for collision and the average number of collisions as follows:

$$D_{C_i} \approx T_C \cdot \frac{\hat{p}_i}{1 - \hat{p}_i} = T_C \cdot \frac{\delta \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j}{1 - (T_S + \delta) \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j}. \quad (19)$$

Furthermore, D_{OC_i} is expressed as the product of the packet collision time and the probability that two or more nodes attempt

transmission at the same time during the backoff process. Consequently, we have

$$
\begin{aligned}
D_{OC_i} = & \frac{D_{B_i}}{\delta} \cdot \left(1 - \prod_{\substack{j=1 \\ j \neq i}}^{N} \left(1 - A\left(\lambda_i, \lambda_j\right) \right) \right) \\
& - \frac{D_{B_i}}{\delta} \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \left[A\left(\lambda_i, \lambda_j\right) \cdot \prod_{\substack{k=1 \\ k \neq j \\ k \neq i}}^{N} \left(1 - A\left(\lambda_i, \lambda_k\right) \right) \right],
\end{aligned}
\tag{20}
$$

where $A\left(\lambda_i, \lambda_j\right) = \frac{\delta \cdot \lambda_j}{1 - \lambda_i \cdot Dos_i}$.

Now, the delay constraint in (14) can be rearranged as follows by combining (16), (18), (19), and (20) all together after removing higher order terms in (20).

$$
\begin{aligned}
& \sum_{j=1}^{N} \lambda_j \cdot T_S + \delta \cdot \frac{CW_{\min} - 1}{2} \cdot \lambda_i \\
& + \left(\delta \cdot \frac{CW_{\min} - 1}{2} + T_C \right) \cdot \frac{\delta \cdot \lambda_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j}{1 - (T_S + \delta) \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j} < 1.
\end{aligned}
\tag{21}
$$

In (21), let $F_{c_i}(\lambda)$ denote $\frac{\delta \cdot \lambda_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j}{1 - (T_S + \delta) \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j}$. Then, we further linearize $F_{c_i}(\lambda)$ at $(\lambda_{\min,1}, \cdots, \lambda_{\min,N})$ as follows:

$$
F_{c_i}(\lambda) \approx F_{c_i}(\lambda_{\min}) + \sum_{j=1}^{N} \frac{\partial F_{c_i}}{\partial \lambda_j}(\lambda_{\min}) \cdot \left(\lambda_j - \lambda_{\min,j}\right),
\tag{22}
$$

where $\lambda_{\min} = \{\lambda_{\min,1}, \ldots, \lambda_{\min,N}\}$. Here, $\frac{\partial F_{c_i}}{\partial \lambda_j}$ is given by

$$
\frac{\partial F_{c_i}}{\partial \lambda_i} = \frac{\delta \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j}{1 - (T_S + \delta) \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j},
\tag{23}
$$

and

$$
\frac{\partial F_{c_i}}{\partial \lambda_k} = \frac{\delta \cdot \lambda_i}{1 - (T_S + \delta) \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j} + \frac{(T_S + \delta) \delta \cdot \lambda_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j}{\left(1 - (T_S + \delta) \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_j \right)^2},
\tag{24}
$$

where $k = 1, \cdots, N$ and $k \neq i$.

Table 1: Parameter values used in performance evaluation.

Notation	Value
T_S	1.094×10^{-4} s
T_C	0.642×10^{-4} s
E_S	0.487×10^{-4} W
E_C	0.286×10^{-4} W
$CW_{\min}$	16
α	5.555
β	1.77
C	0.473×10^{-2}

Consequently, by combining (21), (22), (23), and (24), (21) can be further approximated as

$$
\begin{aligned}
& \sum_{j=1}^{N} \lambda_j T_S + \delta \frac{CW_{\min} - 1}{2} \lambda_i + \left(\delta \frac{CW_{\min} - 1}{2} + T_C \right) \cdot \sum_{j=1}^{N} \frac{\partial F_{c_i}}{\partial \lambda_j}(\lambda_{\min}) \cdot \lambda_j \\
& + \left(\delta \frac{CW_{\min} - 1}{2} + T_C \right) \cdot \left(\frac{\delta \lambda_{\min,i} \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_{\min,j}}{1 - (T_S + \delta) \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N} \lambda_{\min,j}} \right. \\
& \left. - \sum_{j=1}^{N} \frac{\partial F_{c_i}}{\partial \lambda_j}(\lambda_{\min}) \cdot \lambda_{\min,j} \right) < 1.
\end{aligned}
\tag{25}
$$

It is straightforward to show the convexity of (25).

Finally, since the energy consumption of the objective function in (3) is not in a closed form, we further approximate the objective function by replacing p_i with $\hat{p}_i$ as follows:

$$
E_{\text{tot}} \approx \sum_{i=1}^{N} \left[\lambda_i \left(E_S + F_{c_i}(\lambda) \cdot E_C \right) \right].
\tag{26}
$$

By using (25) and (26), the optimization problem in (3) can be relaxed as convex optimization. Then, the optimal solution for the relaxed convex problem can be found in polynomial time by using the interior point method, which will be further discussed in Section 4.2.

4 PERFORMANCE EVALUATION

4.1 Testbed

In order to create a testbed for wireless control systems, we construct 20 disk-levitation tubes, each of which wirelessly controls the levitating height of a disk as shown in Fig. 2. Each disk-levitation tube consists of a DC motor with a propeller, a lightweight disk, an infrared sensor for measuring the height of the disk, and two computing boards. The control algorithm for levitating the disk to the setpoint is implemented in the PC. The NodeMCU board, which contains the ESP8266 Wi-Fi module, transmits the measurement data to the PC. We use the Arduino UNO board for operating the motor and the infrared sensor. Two boards transmit the data via serial communication.

Figure 2: The disk-levitation tube testbed, which controls the height of 20 disks at the same time.

Table 2: Comparison of the solutions in Fig. 3.

(W, w_c)	Greedy: $(\lambda_1^*, \lambda_2^*, F^*)$	SA: $(\lambda_1^*, \lambda_2^*, F^*)$	Interior point: $(\lambda_1^*, \lambda_2^*, F^*)$	Optimum: $(\lambda_1^*, \lambda_2^*, F^*)$
$(0.2, [0.1\ 0.4])$	$(0.20, 0.17, 0.00103291)$	$(0.20773, 0.16006, 0.00103029)$	$(0.20734, 0.16096, 0.00103027)$	$(0.20733, 0.16096, 0.00103027)$
$(0.2, [0.25\ 0.25])$	$(0.17, 0.17, 0.00104926)$	$(0.17551, 0.17565, 0.00104783)$	$(0.17624, 0.17624, 0.00104781)$	$(0.17621, 0.17625, 0.00104781)$
$(0.2, [0.25\ 0.75])$	$(0.17, 0.14, 0.00161248)$	$(0.17655, 0.14159, 0.00161169)$	$(0.17620, 0.14144, 0.00161169)$	$(0.17625, 0.14149, 0.00161169)$
$(0.5, [0.1\ 0.4])$	$(0.17, 0.11, 0.00168154)$	$(0.16315, 0.11982, 0.00167768)$	$(0.16096, 0.11938, 0.00167760)$	$(0.16095, 0.11942, 0.00167760)$
$(0.5, [0.25\ 0.25])$	$(0.14, 0.14, 0.00169708)$	$(0.13273, 0.13293, 0.00169469)$	$(0.13297, 0.13297, 0.00169469)$	$(0.13283, 0.13283, 0.00169469)$
$(0.5, [0.25\ 0.75])$	$(0.14, 0.11, 0.00298518)$	$(0.13264, 0.10176, 0.00298146)$	$(0.13287, 0.10241, 0.00298144)$	$(0.13284, 0.10246, 0.00298144)$
$(0.8, [0.1\ 0.4])$	$(0.11, 0.08, 0.00220088)$	$(0.11940, 0.08471, 0.00219940)$	$(0.11941, 0.08371, 0.00219938)$	$(0.11940, 0.08371, 0.00219938)$
$(0.8, [0.25\ 0.25])$	$(0.11, 0.11, 0.00222001)$	$(0.09502, 0.09561, 0.00221065)$	$(0.09524, 0.09524, 0.00221065)$	$(0.09512, 0.09511, 0.00221065)$
$(0.8, [0.25\ 0.75])$	$(0.11, 0.08, 0.00419790)$	$(0.09543, 0.07049, 0.00418851)$	$(0.09529, 0.06980, 0.00418849)$	$(0.09513, 0.06979, 0.00418849)$

We obtain a dynamic model of the tube by measuring the motion of the disk when the step input is injected into the motor. The derived transfer function from motor command to the disk height is $H(s) = 0.6/(s + 0.2)$. We design a PI controller with the proportional gain $k_P = 0.5$ and integral gain $k_I = 0.1$. The setpoint of the disk height for every tube is given by 0.4 meters.

4.2 Simulation Study

In this section, we carry out simulation study to show the effectiveness of the proposed optimization framework. The parameter values are summarized in Table 1. It should be noted that these values are from the data sheet of the Wi-Fi ESP8266 chip used in our testbed for empirical study in the subsequent section [1].

The first aim of the simulation study is to validate the convex relaxation approach. In this regard, we use two general algorithms, greedy heuristic and simulated annealing, to solve (3). Then, we compare the solutions with the one obtained by the convex relaxation approach. The greedy heuristic algorithm searches for the sampling period λ that minimizes the objective function by decreasing each component of λ by a step size of μ in an iterative manner starting from its maximum value. Simulated annealing is a general probability based algorithm for approximating the global optimum

of the optimization problem. This technique gives a good approximation to the global optimum solution of a given function in a vast search space. Finally, for the relaxed optimization, we adopt the interior point method for solving convex optimization problems. The inequality constraint of the convex optimization problem can be encoded using the logarithmic barrier function. The interior point method provably converges to the optimal solution in polynomial time.

In the simulation study, we consider the wireless network control system with two disk levitation tubes. Then, we have $\lambda = (\lambda_1, \lambda_2)$. By using the parameter values in Table 1, we plot the objective function surface satisfying the constraints in (3) for various cases of weight assignments in Fig. 3 using Matlab. We further plot the solutions obtained by the three algorithms, respectively. From Fig. 3, it is clear that the solutions of three algorithms are very close each other. The solutions are summarized in Table 2, which validates that all the three algorithms show similar performance that is very close to the optimal. We can conclude that the relaxed convex optimization gives very promising solutions in the case of two control loops.
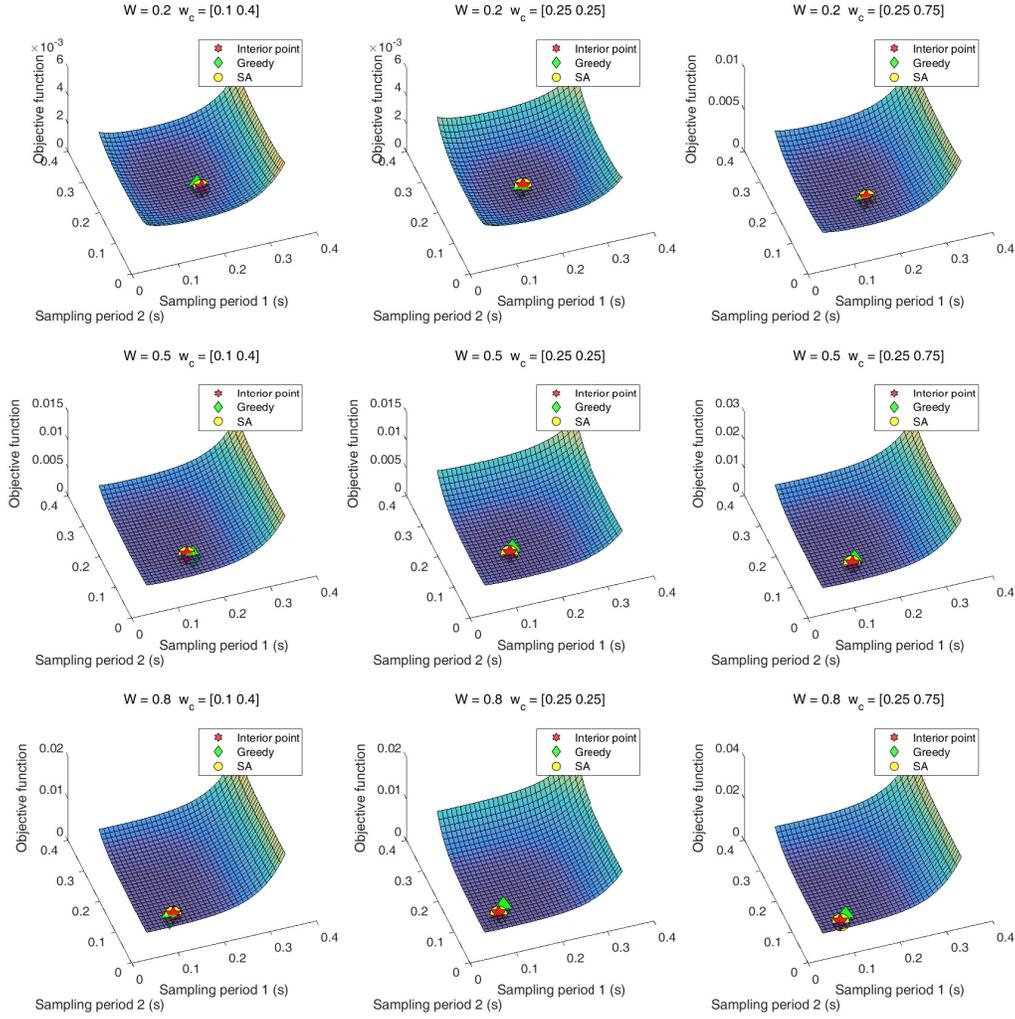
Figure 3: Objective function surface and solutions of three algorithms when there are two control loops.

The second aim is to show the effectiveness of the co-design approach. Clearly, the solution $\lambda = (\lambda_1, \lambda_2)$ obtained by the co-design method accomplishes the smallest value of the objective function in all 9 cases shown in Fig. 3.

Since the solutions of the three algorithms are quite close to each other in Fig. 3, we check the computation time of the algorithms in Fig. 4, which shows the computation time with respect to the number of nodes. In the left of Fig. 4, the computation time of SA significantly increases while those of others remain small. The right of Fig. 4 shows further comparison between the interior point method and greedy heuristic, which confirms that the interior point method is much superior to the greedy heuristic in terms of the computation time.

Now, we consider the optimization performance when there are $N = 10$ nodes. Unlike the two-node case, it is impossible to visualize the surface that requires an $(N + 1)$ dimensional space. Hence, for the baseline, we plot the homogeneous case when all the nodes have the same sampling period. It should be noted that the x-axis only applies to the homogeneous case. In Fig. 5, we can notice that the performance gap between the homogeneous case and the three algorithms becomes larger with unbalanced weights in the objective function, which is obvious because the search space of the homogeneous problem is limited.

Again, for the 10 node case, the co-design method determines the combination of 10 sampling periods with which the objective
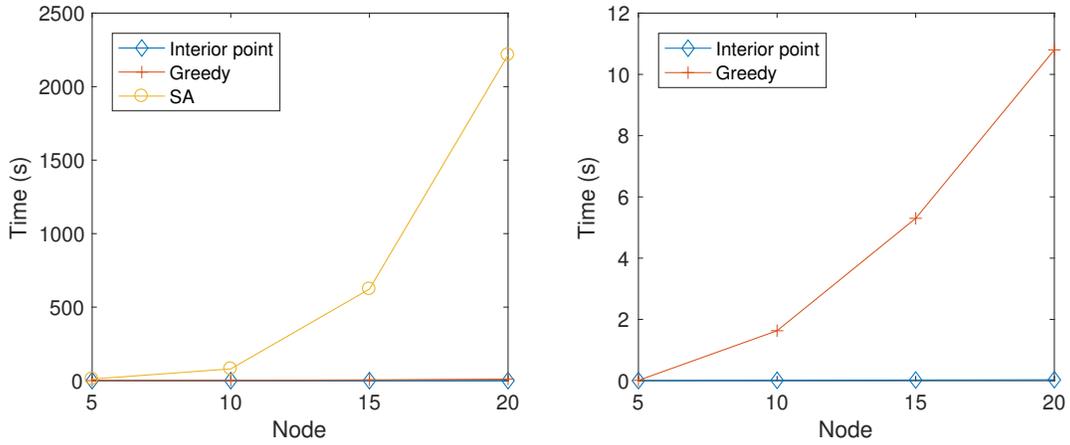
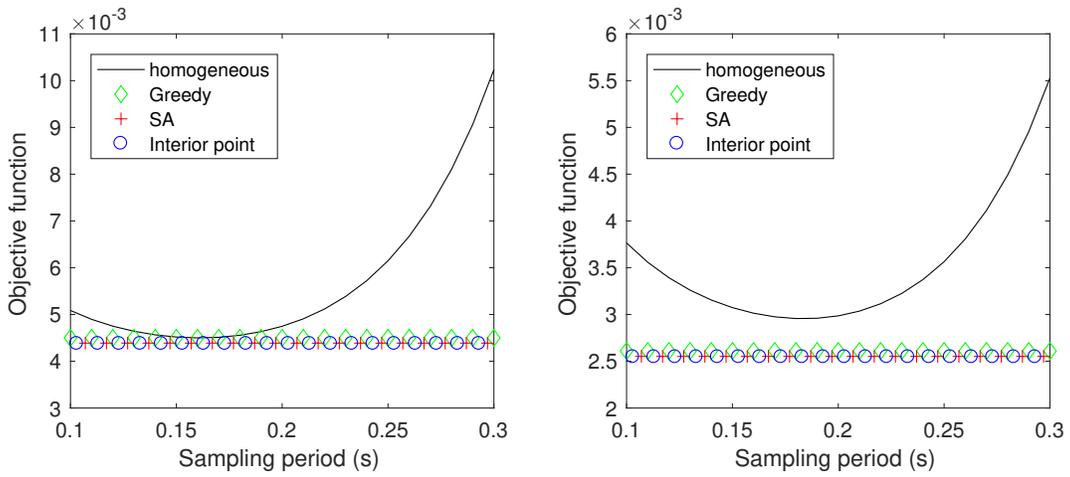Figure 4: Computation time of the three algorithms.

Figure 5: Objective function with respect to the sampling period when all the nodes have the same sampling period (left: $N = 10$, $W = 0.5$, $w_c = 1/N$, right: $N = 10$, $W = 0.5$, $w_{c_{1,\ldots,N/2}} = 1/N$, $w_{c_{N/2+1,\ldots,N}} = 0.001/N$.)

function achieves a smaller value compared to any values obtained in the homogeneous cases.

4.3 Experiments

The experiments are carried out in the testbed. The control cost of J_i of the i-th tube is defined by the accumulated absolute value of the disk height deviation from the setpoint for 100 seconds divided by the number of measurement samples, then multiplied by the weight w_{c_i}. The energy consumption E_{tot} is calculated by multiplying E_C and E_S by counting the number of collisions and the number of packets sent by all nodes for 100 seconds, then dividing it by 100.

Figure 6 shows the value of the empirical objective function with respect to the number of nodes and W, respectively. The sampling periods of all the nodes are the same value. The blue points show the minimum values of the objective function with respect to the sampling period for a fixed value of the number of nodes and W, respectively. Figure 7 compares the empirical objective function

obtained by the interior point method and that in Fig. 6 for evaluating the performance with the optimization. It should be noted that each value of the empirical objective function by the interior point method is an experimental result of the testbed by setting λ to the solution by the interior point method. The results in Fig. 7 indicate that the objective function value can be obtained better than the homogeneous case when the co-design method is applied.

In a similar manner, Figure 8 shows the value of the empirical objective function with respect to the number of nodes and W, respectively, when the weights of the control cost, w_c, are significantly unbalanced. Similarly as the blue points in Fig. 6, those in Fig. 8 indicates the points where the objective function achieves the minimum with respect to the sampling period for each fixed value of the number of nodes and W, respectively. Figure 9 shows the difference between the objective function by the interior point method and the homogeneous case. As the weights w_C become more unbalanced, the solution by the interior point method gets far
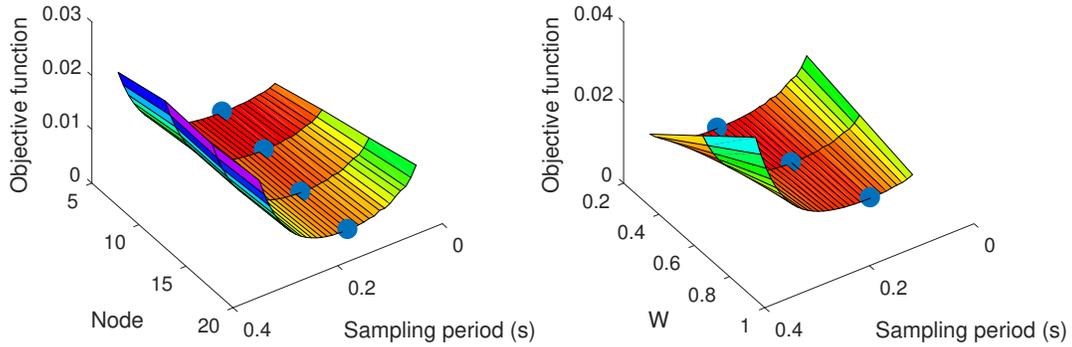
Figure 6: Empirical results of the objective function with respect to the homogeneous sampling period (left: $N = [5, 10, 15, 20]$**,** $W = 0.5$**,** $w_{c_{1,\ldots,N/2}} = 1.5/N$**,** $w_{c_{N/2+1,\ldots,N}} = 0.5/N$ **right:** $N = 15$**,** $W = [0.2, 0.5, 0.8]$**,** $w_{c_{1,\ldots,N/2}} = 1.5/N$**,** $w_{c_{N/2+1,\ldots,N}} = 0.5/N$**.)**
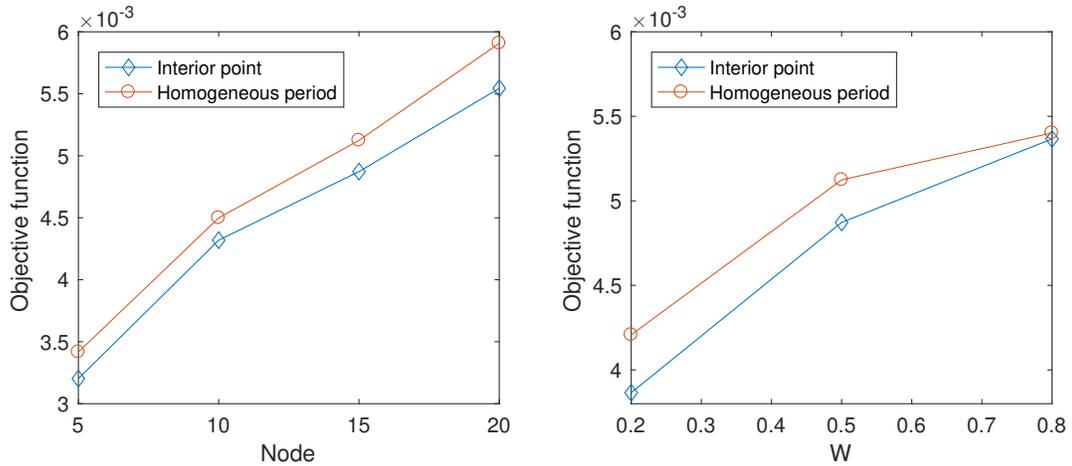
Figure 7: Comparison of the empirical performance between the homogeneous case in Fig. 6 and that of the interior point method (left: $N = [5, 10, 15, 20]$**,** $W = 0.5$**,** $w_{c_{1,\ldots,N/2}} = 1.5/N$**,** $w_{c_{N/2+1,\ldots,N}} = 0.5/N$ **right:** $N = 15$**,** $W = [0.2, 0.5, 0.8]$**,** $w_{c_{1,\ldots,N/2}} = 1.5/N$**,** $w_{c_{N/2+1,\ldots,N}} = 0.5/N$**.)**
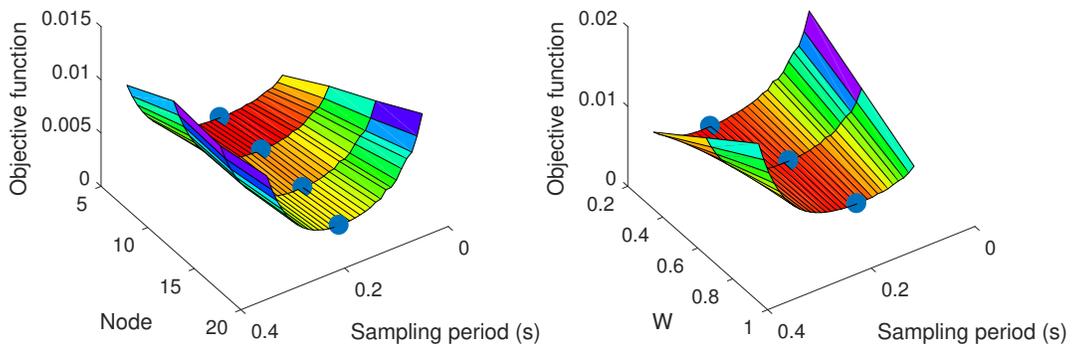
Figure 8: Empirical results of the objective function with respect to the homogeneous sampling period with significant difference between w_c **(left:** $N = [5, 10, 15, 20]$**,** $W = 0.5$**,** $w_{c_{1,\ldots,N/2}} = 1/N$**,** $w_{c_{N/2+1,\ldots,N}} = 0.001/N$**, right:** $N = 15$**,** $W = [0.2, 0.5, 0.8]$**,** $w_{c_{1,\ldots,N/2}} = 1/N$**,** $w_{c_{N/2+1,\ldots,N}} = 0.001/N$**.)**
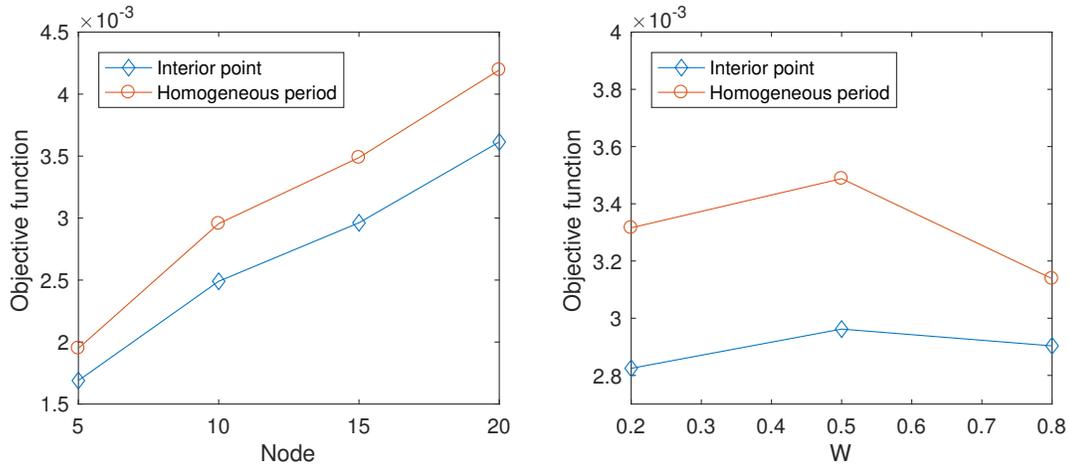
Figure 9: Comparison of the empirical performance between the homogeneous case in Fig. 8 and that of the interior point method (left: $N = [5, 10, 15, 20]$, $W = 0.5$, $w_{c_{1,\ldots,N/2}} = 1/N$, $w_{c_{N/2+1,\ldots,N}} = 0.001/N$, **right:** $N = 15$, $W = [0.2, 0.5, 0.8]$, $w_{c_{1,\ldots,N/2}} = 1/N$, $w_{c_{N/2+1,\ldots,N}} = 0.001/N$.**)**

away from the homogeneous case. As expected, the performance difference becomes larger in Fig. 9 than those in Fig. 7.

5 CONCLUSIONS AND FUTURE WORK

We have formulated an optimization framework for wireless control systems, where the objective function is a weighted sum of the control cost and the energy consumption for transmission while the constraint is to satisfy the network delay requirements. In our formulation, the control variables are the sampling periods of all the feedback loops. We have shown the effectiveness of the proposed optimization framework with extensive simulation studies as well as experimental results based on a testbed.

There are several directions for future work. One is to develop a real-time adaptive architecture for random access network and control co-design. Another direction is to develop a joint optimization framework for network parameters and sampling periods.

ACKNOWLEDGMENTS

REFERENCES

[1] 2018. ESP8266EX Datasheet. Retrieved Oct 16, 2018 from https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
[2] Jia Bai, Emeka P Eyisi, Fan Qiu, Yuan Xue, and Xenofon D Koutsoukos. 2012. Optimal cross-layer design of sampling rate adaptation and network scheduling for wireless networked control systems. In *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems.* 107–116.
[3] Terrence Blevins, Deji Chen, Mark Nixon, and Willy Wojsznis. 2015. *Wireless Control Foundation: Continuous and Discrete Control for the Process Industry.* Vol. 4. International Society of Automation.
[4] Simone Brienza, Manuel Roveri, Domenico De Guglielmo, and Giuseppe Anastasi. 2016. Just-in-time adaptive algorithm for optimal parameter setting in 802.15. 4 WSNs. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 10, 4 (2016), 27.
[5] Burak Demirel, Zhenhua Zou, Pablo Soldati, and Mikael Johansson. 2014. Modular design of jointly optimal controllers and forwarding policies for wireless control.

IEEE Trans. Automat. Control 59, 12 (2014), 3252–3265.
[6] Konstantinos Gatsis, Alejandro Ribeiro, and George J Pappas. 2016. Control-aware random access communication. In *Cyber-Physical Systems (ICCPS), 2016 ACM/IEEE 7th International Conference on.* IEEE, 1–9.
[7] Kyoung-Dae Kim and Panganamala R Kumar. 2012. Cyber–physical systems: A perspective at the centennial. *Proc. IEEE* 100, Special Centennial Issue (2012), 1287–1308.
[8] Bo Li, Yehan Ma, Tyler Westenbroek, Chengjie Wu, Humberto Gonzalez, and Chenyang Lu. 2016. Wireless routing and control: a cyber-physical case study. In *Cyber-Physical Systems (ICCPS), 2016 ACM/IEEE 7th International Conference on.* 1–10.
[9] Chenyang Lu, Abusayeed Saifullah, Bo Li, Mo Sha, Humberto Gonzalez, Dolvara Gunatilaka, Chengjie Wu, Lanshun Nie, and Yixin Chen. 2016. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proc. IEEE* 104, 5 (2016), 1013–1024.
[10] Kyung-Joon Park, Rong Zheng, and Xue Liu. 2012. Cyber-physical systems: Milestones and research challenges. *Computer Communications* 36, 1 (2012), 1–7.
[11] Pangun Park, José Araújo, and Karl Henrik Johansson. 2011. Wireless networked control system co-design. In *Networking, Sensing and Control (ICNSC), 2011 IEEE International Conference on.* IEEE, 486–491.
[12] Pangun Park, Piergiuseppe Di Marco, Carlo Fischione, and Karl Henrik Johansson. 2013. Modeling and optimization of the IEEE 802.15. 4 protocol for reliable and timely communications. *IEEE Transactions on Parallel and Distributed Systems* 24, 3 (2013), 550–564.
[13] Pangun Park, Piergiuseppe Di Marco, and Karl Henrik Johansson. 2017. Cross-layer optimization for industrial control applications using wireless sensor and actuator mesh networks. *IEEE Transactions on Industrial Electronics* 64, 4 (2017), 3250–3259.
[14] Pangun Park, Sinem Coleri Ergen, Carlo Fischione, Chenyang Lu, and Karl Henrik Johansson. 2018. Wireless network design for control systems: A survey. *IEEE Communications Surveys & Tutorials* 20, 2 (2018), 978–1013.
[15] Pangun Park, Sinem Coleri Ergen, Carlo Fischione, and Alberto Sangiovanni-Vincentelli. 2013. Duty-cycle optimization for IEEE 802.15. 4 wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 10, 1 (2013), 12.
[16] Yalcin Sadi, Sinem Coleri Ergen, and Pangun Park. 2014. Minimum energy data transmission for wireless networked control systems. *IEEE Transactions on Wireless Communications* 13, 4 (2014), 2163–2175.
[17] Abusayeed Saifullah, Chengjie Wu, Paras Babu Tiwari, You Xu, Yong Fu, Chenyang Lu, and Yixin Chen. 2014. Near optimal rate selection for wireless control systems. *ACM Transactions on Embedded Computing Systems (TECS)* 13, 4s (2014), 128.
[18] Omesh Tickoo and Biplab Sikdar. 2008. Modeling queueing and channel access delay in unsaturated IEEE 802.11 random access MAC based wireless networks. *IEEE/ACM Transactions on Networking (TON)* 16, 4 (2008), 878–891.
[19] Yuchang Won, Buyeon Yu, Jaegeun Park, In-Hee Park, Haegeon Jeong, Jeanseong Baik, Kyungtae Kang, Insup Lee, Sang Hyuk Son, Kyung-Joon Park, and Yongsoon Eun. 2018. An attack-resilient CPS architecture for hierarchical control: A case study on train control systems. *Computer* 51, 11 (2018), 46–55.